

AMENDMENTS TO THE CLAIMS

Cancel claims 4-8, 10-12, 14-19, 21, 23-32, 37-40 and 46-58 without prejudice.

This listing of claims will replace all prior versions, and listings, of claims in the application:

1-58. (canceled)

59. (currently amended) ~~The method of claim 54~~

A method in a computer system for dispatching requests to perform services to sub-applications that use different logic models the method comprising:

providing a context for the sub-applications

receiving a request from a client computer to perform a service; and

for a plurality of sub-applications,

determining whether the received request should be dispatched to the sub-application; and

when it is determined that the request should be dispatched to the sub-application, invoking a service routine of the sub-application passing the request

whereby the sub-applications share the provided context;

wherein the determining includes determining whether a match criteria for the sub-application matches the received request;

wherein the requests are HTTP requests with a URL and the match criteria is a regular expression relating to the URL;

wherein a respective service routine is invoked for the request with respect to each of at least two of the sub-applications; and

wherein the sub-applications are ordered and the invoking of the service routines of the at least two sub-applications is performed in the order of the sub-applications.

60. (currently amended) ~~The computer system of claim 55~~

A computer system for dispatching HTTP requests to sub-applications,
comprising:
a configuration file having a class, initialization parameters, and a match criteria
associated with the sub-applications;
an initialization component that instantiates an object of the class for each sub-
application in the configuration file, the instantiated object being initialized with the initialization
parameters for the sub-application and being provided with a context object, the context object
being shared by the instantiated objects so that the sub-applications share a common context; and
a dispatcher that receives HTTP requests from client computers and, when the
received HTTP request matches a match criteria of a sub-application, invokes a service routine of
the instantiated object of the class associated with the sub-application;
wherein the match criteria is a regular expression relating to a URL of the HTTP
request;
wherein a respective service routine is invoked for at least one of the HTTP requests with
respect to each of at least two of the sub-applications; and
wherein the configuration file specifies an ordering of the sub-applications and the
dispatcher invokes the service routines of the instantiated objects of the classes associated with
the at least two sub-applications in the specified order.

61. (currently amended) ~~The computer system of claim 56~~

A computer system for processing request messages, comprising:
a plurality of sub-applications forming an application, a sub-application having a
match criteria indicating when the sub-application should process a request and having a service
routine to invoke when the match criteria indicates that the sub-application should process the
request, the sub-applications using disparate logic models;
a context for the application that is shared by the sub-applications; and
a dispatcher that receives requests from client computers, evaluates the match
criteria to identify which sub-applications have match criteria that match the requests, and

invokes the service routines of the identified sub-applications wherein invoked sub-applications use the context;

wherein the requests are HTTP requests with a URL and the match criteria is a regular expression relating to the URL;

wherein a respective service routine is invoked for at least one of the requests with respect to each of at least two of the sub-applications; and

wherein the sub-applications are ordered and the dispatcher invokes the service routines of the at least two sub-applications based on the order of the sub-applications.

62. (previously presented) The computer system of claim 61 wherein an invoked service routine indicates that additional service routines should not be invoked to process the received request.

63. (previously presented) The computer system of claim 61 wherein the dispatcher does not invoke additional service routines when an invoked service routine responds to a received request.

64. (currently amended) ~~The computer system of claim 57~~

A computer system for processing request messages, comprising:

a plurality of service means for servicing requests, the service means forming an application, each service means having a match criteria indicating when the service means should be invoked, the service means implementing different logic models; and

dispatch means for receiving requests from client computers, evaluating match criteria to identify which service means have match criteria that match the requests, and invoking the identified service means whereby the service means share a context that is common to the service means of the application;

wherein the requests are HTTP requests with a URL and the match criteria is a regular expression relating to the URL;

wherein at least one of the requests is serviced by at least two of the service means; and
wherein the service means are ordered and the dispatch means invokes the at least two service means based on their order.

65. (previously presented) The computer system of claim 64 wherein an invoked service means indicates that additional service means should not be invoked to process the received request.

66. (previously presented) The computer system of claim 64 wherein the dispatch means does not invoke additional service means when an invoked service means responds to a received request.

67. (currently amended) ~~The computer-readable medium of claim 58~~

A computer-readable medium for controlling a computer system to dispatch requests to perform services to service routines, by a method comprising:
receiving a request from a client computer to perform a service; and
for a plurality of service routines,
retrieving a match criteria for the service routine;
determining whether the received request matches the retrieved match
criteria;
when it is determined that the request matches the retrieved match criteria,
invoking the service routine for processing of the received request;
whereby the service routines form an application and share a common context;
wherein the requests are HTTP requests with a URL and the match criteria is a regular expression relating to the URL;
wherein at least one of the requests is processed by at least two of the service routines;
and
wherein the service routines are ordered and the invoking of the at least two service means is performed in the order of the service routines.